# $r$-indexing Wheeler graphs

Travis Gagie
Dalhousie University

DSB 2021
12/02/2021*

*Palindrome! :o)

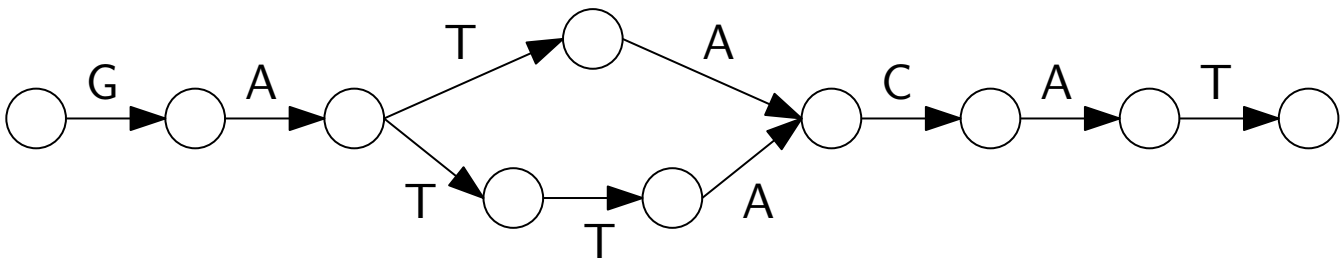
# Wheeler graphs

# definition

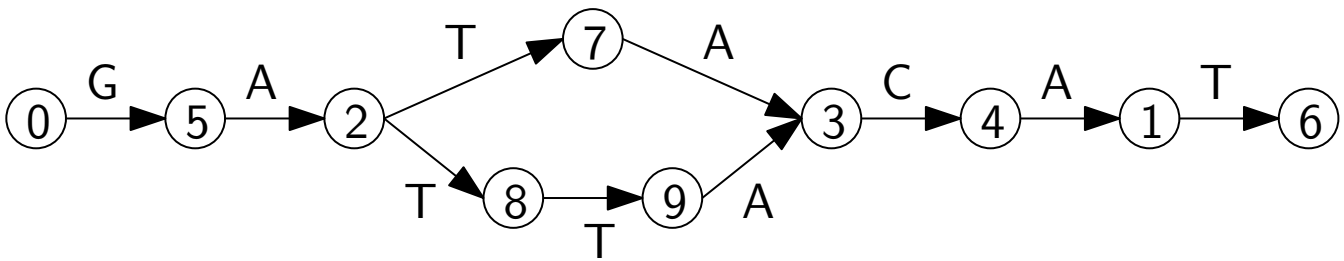
A Wheeler graph is a directed edge-labelled graph whose vertices can be ordered such that those with in-degree 0 precede those with positive in-degree and, for any pair of edges $e = (u, v)$ and $e' = (u', v')$ labelled $a$ and $a'$ respectively,

- if $a \prec a'$ then $v < v'$,
- if $a = a'$ and $u < u'$ then $v \leq v'$.

# example



# example

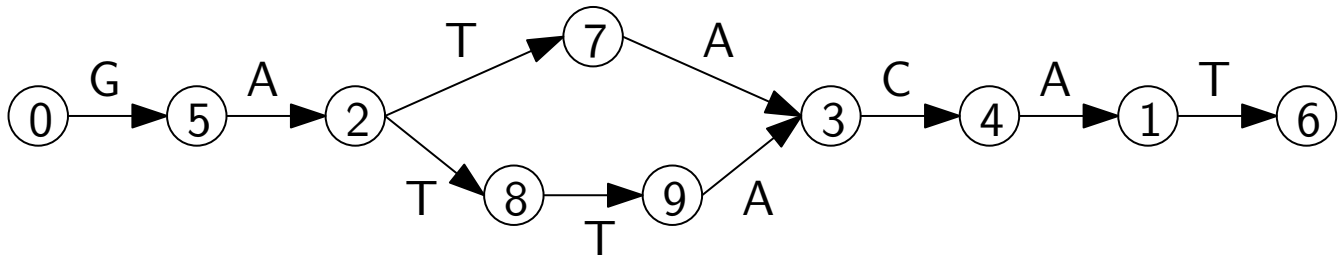


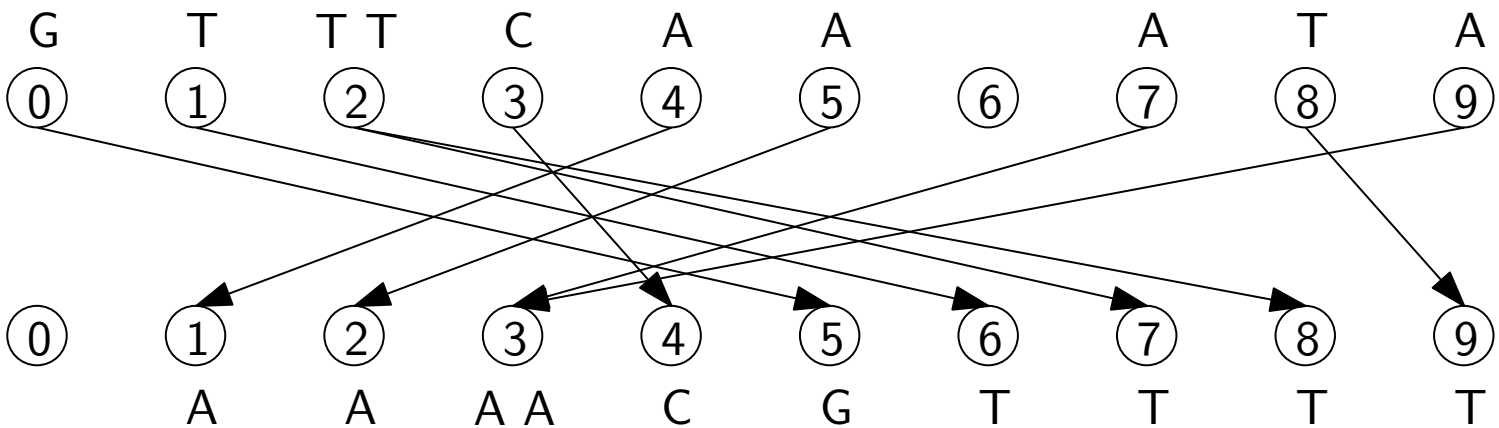
# result

**Theorem**

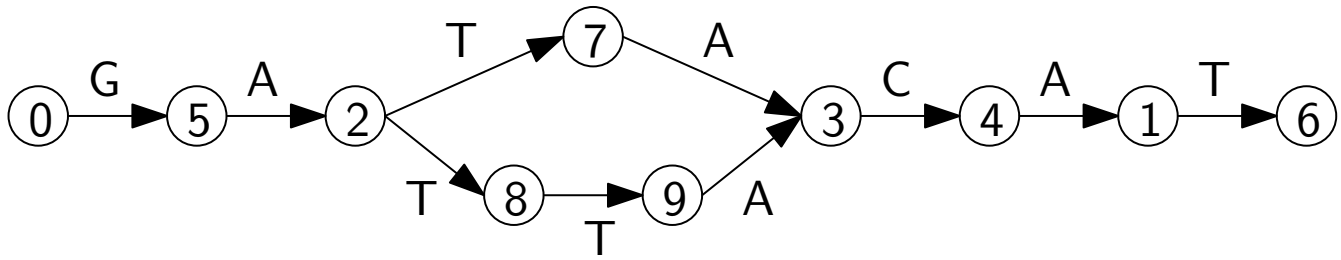
*If $G$ is a Wheeler graph and we store a rank data structure over its edge-labels, ordered by their origins, and partial sum data structures over its vertices' in- and out-degrees then later, given a pattern $P$, in $O(|P| \log\log |G|)$ time we can find the interval in the Wheeler order containing vertices reachable by paths labelled $P$.*

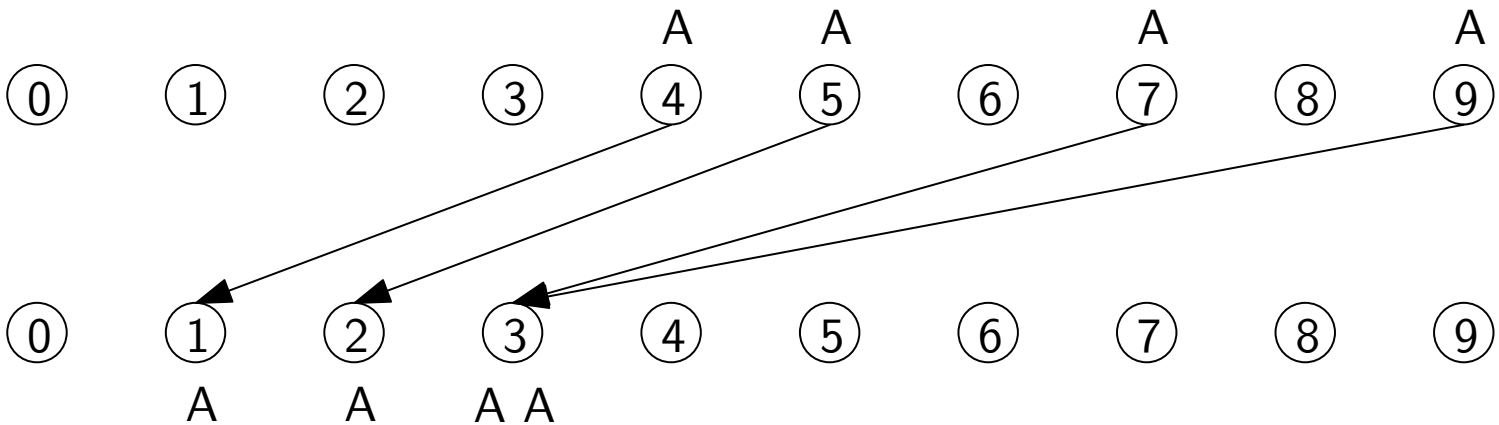
# PbID



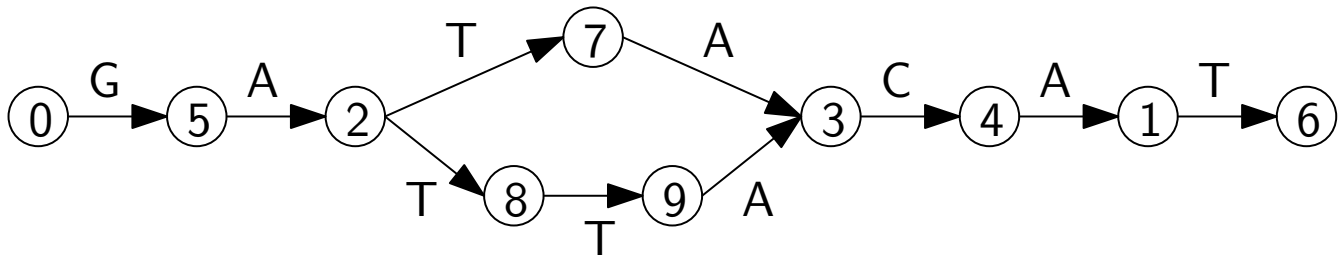



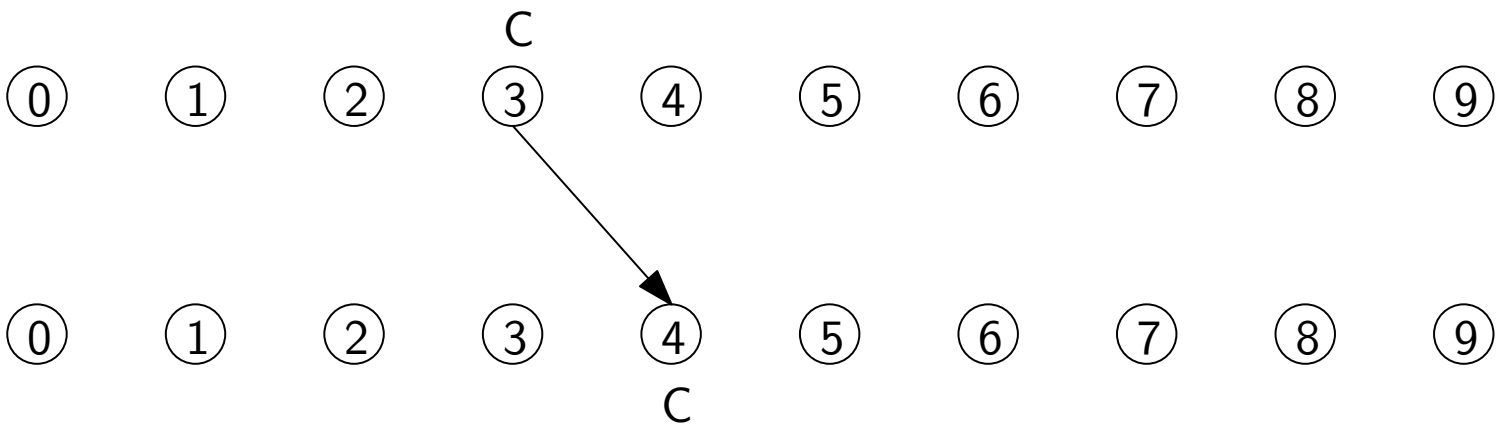
# PbID



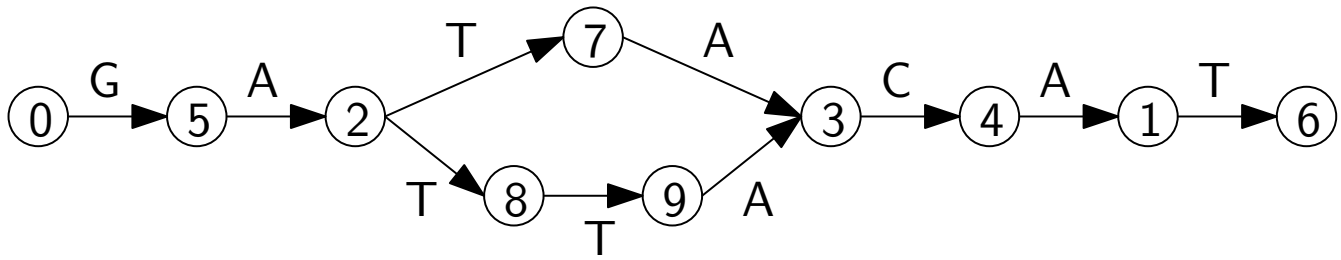



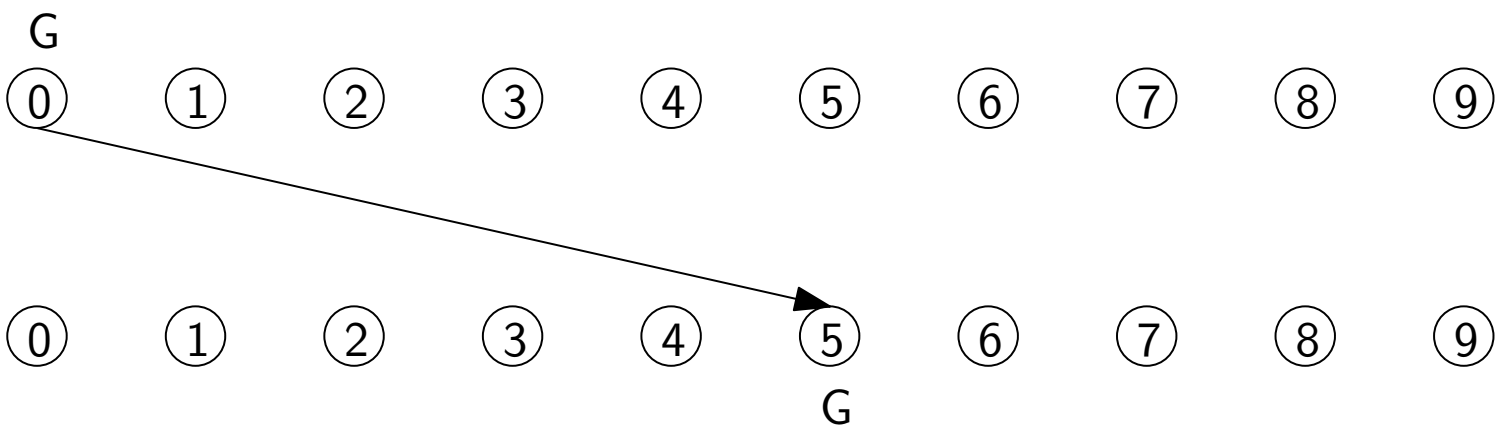
# PbID



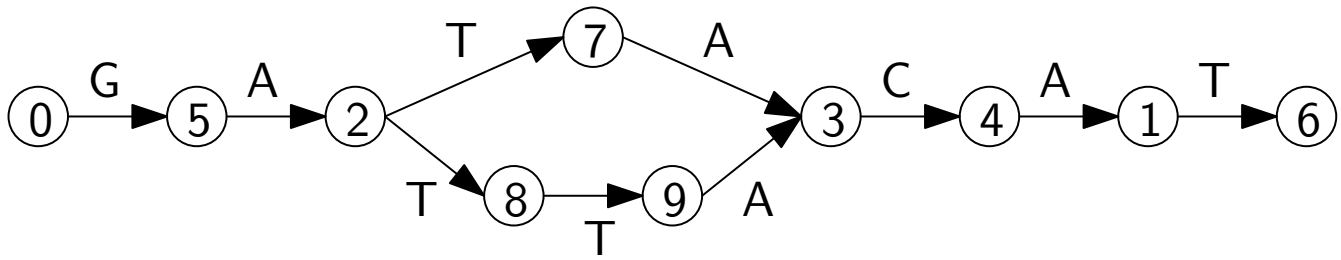



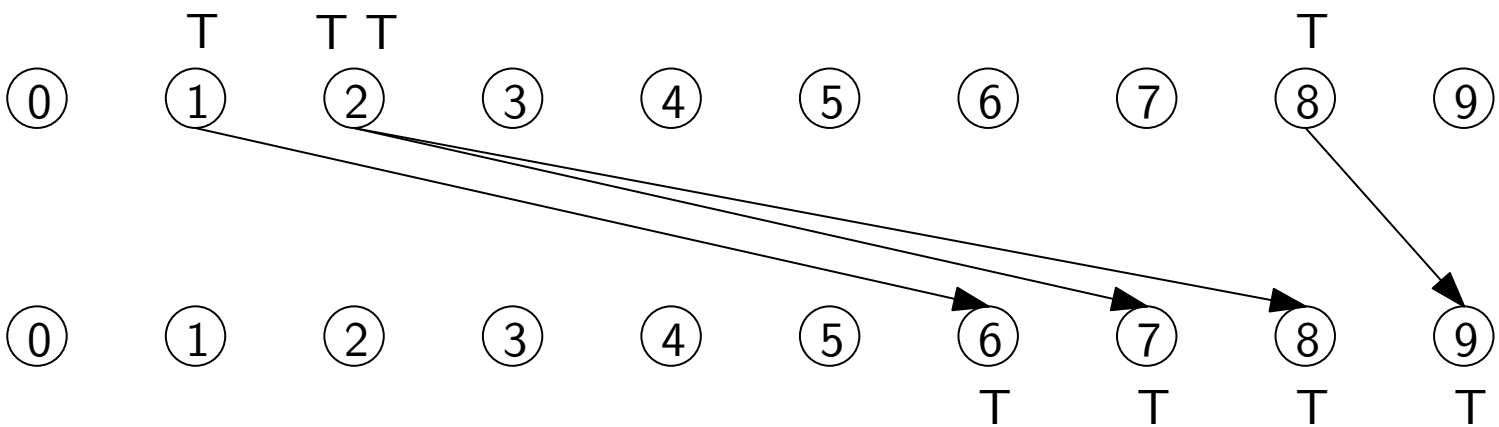
# PbID



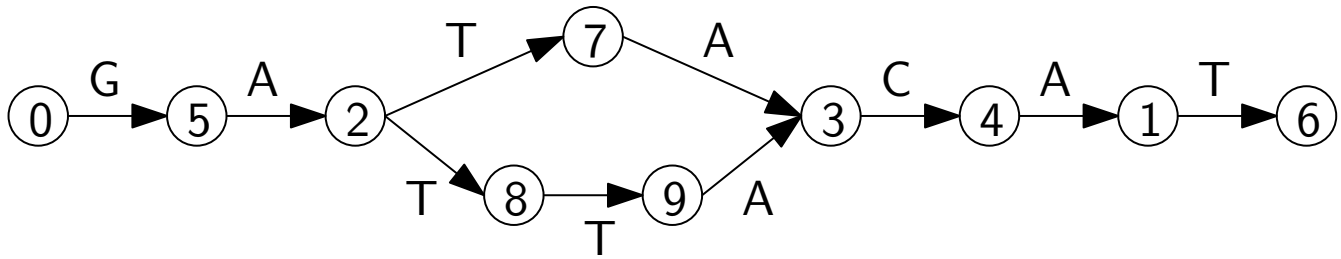
# PbID

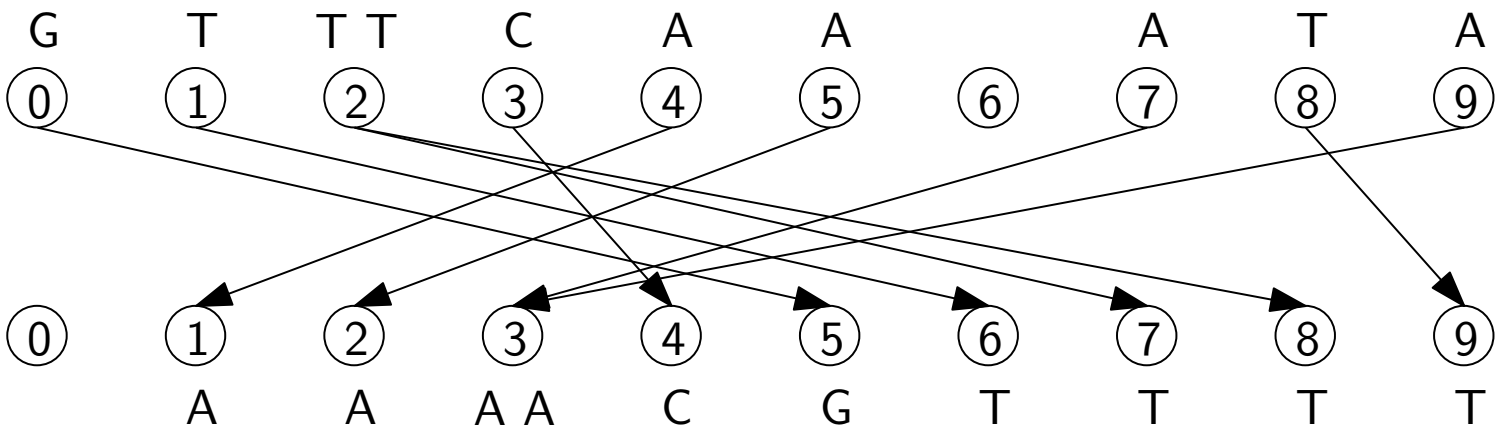


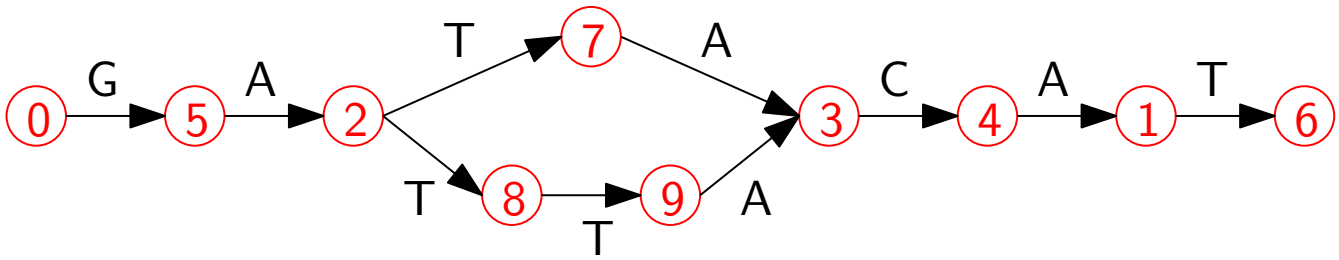
# mnemonic

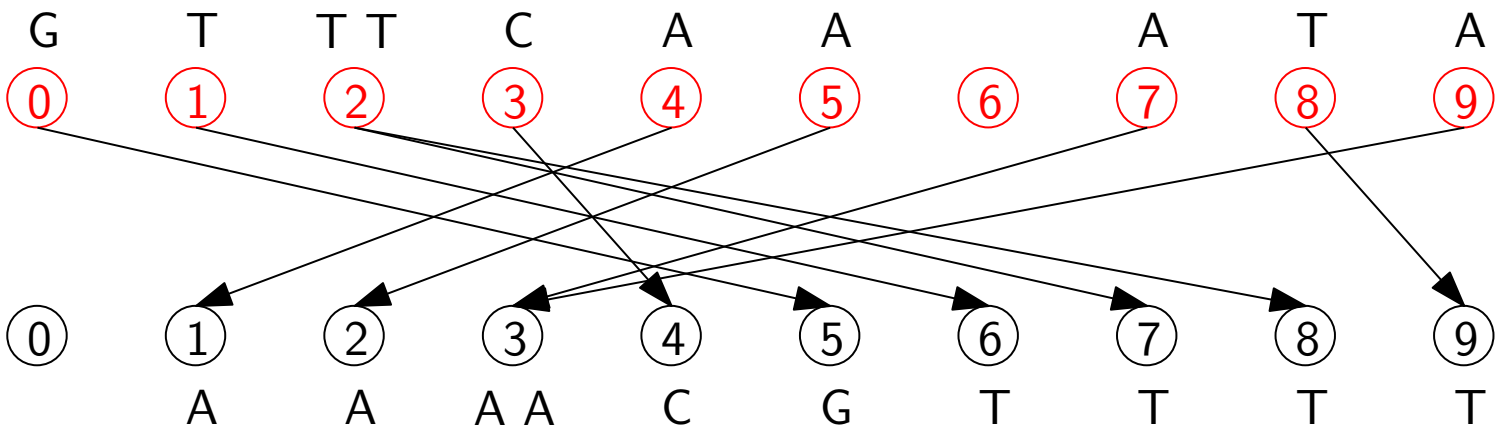
# seeking TA



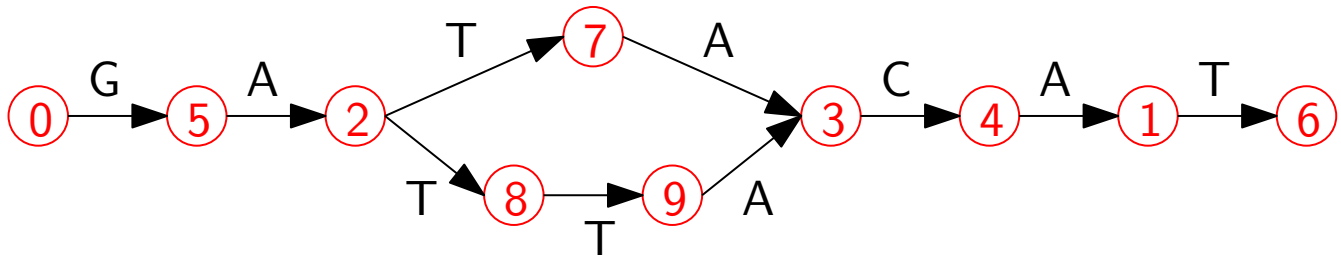



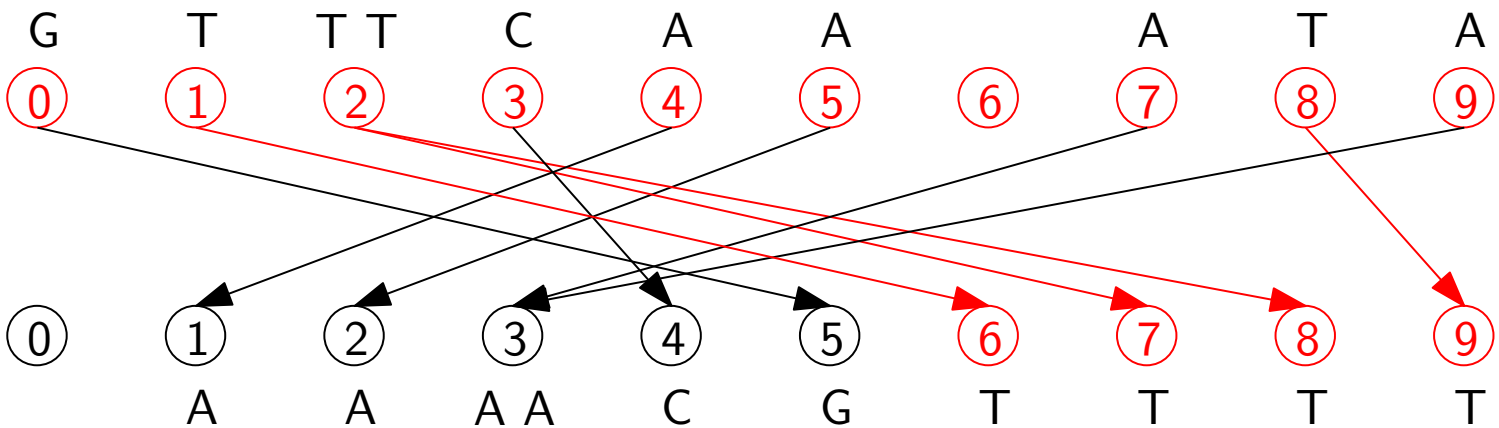
# seeking TA



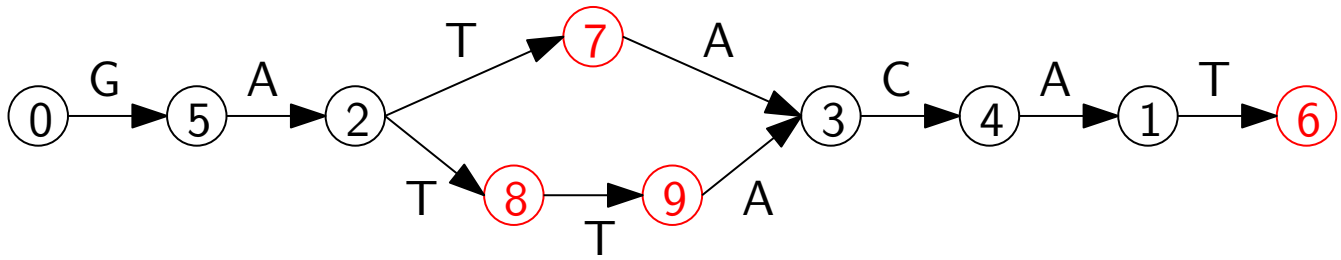



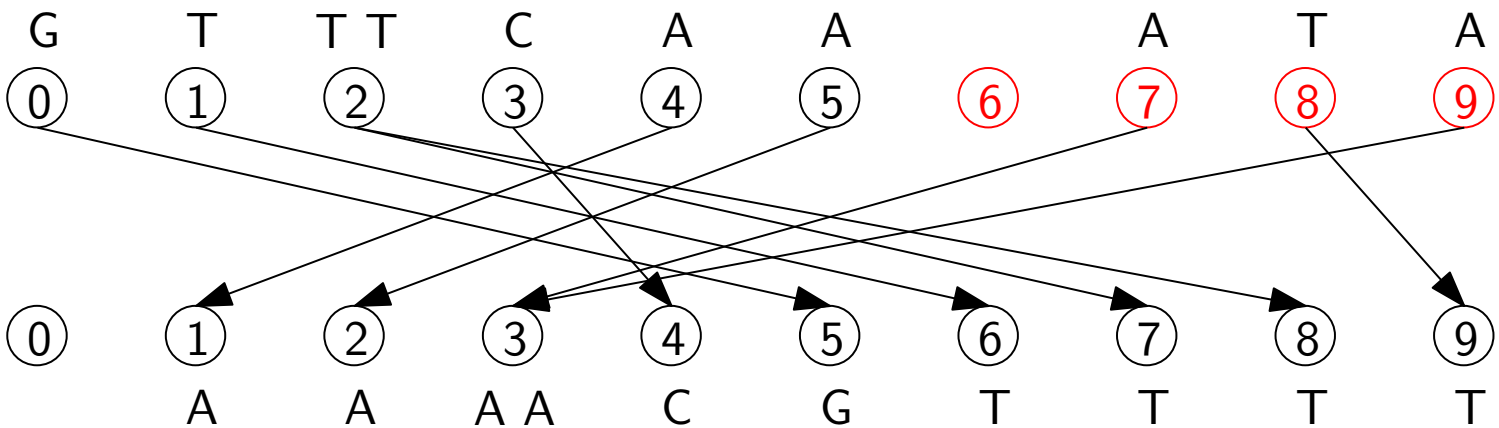
# seeking TA



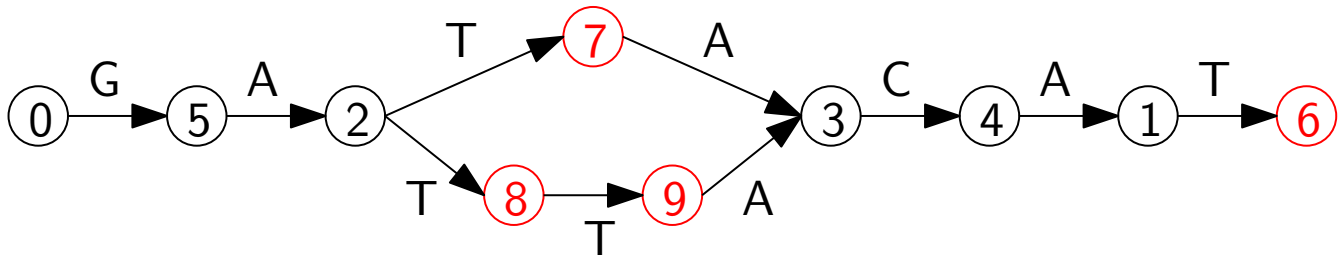



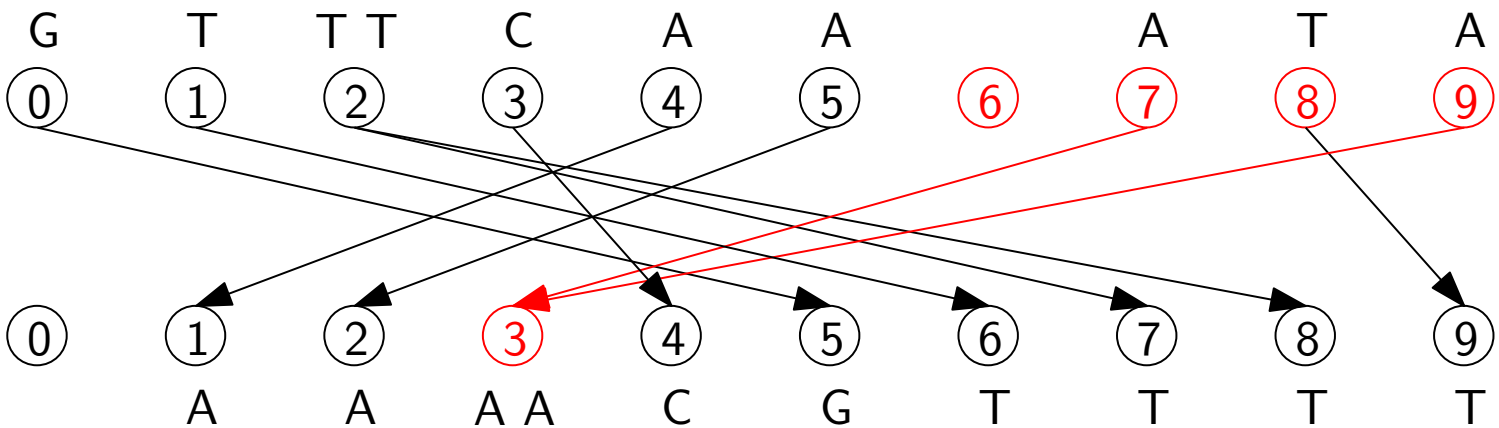
# seeking TA



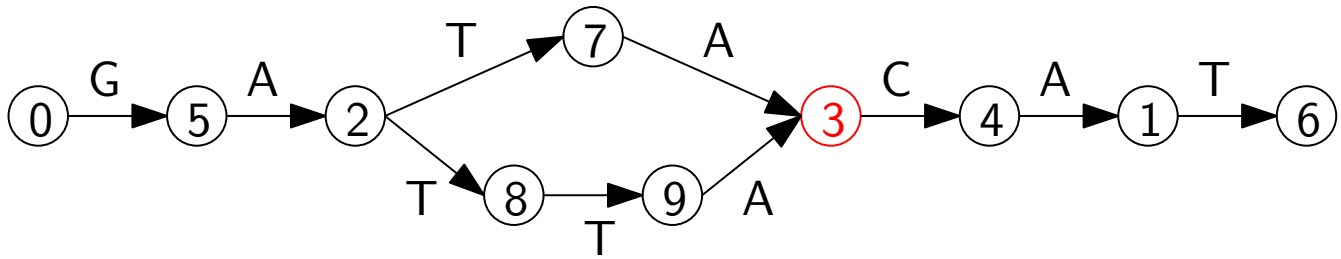



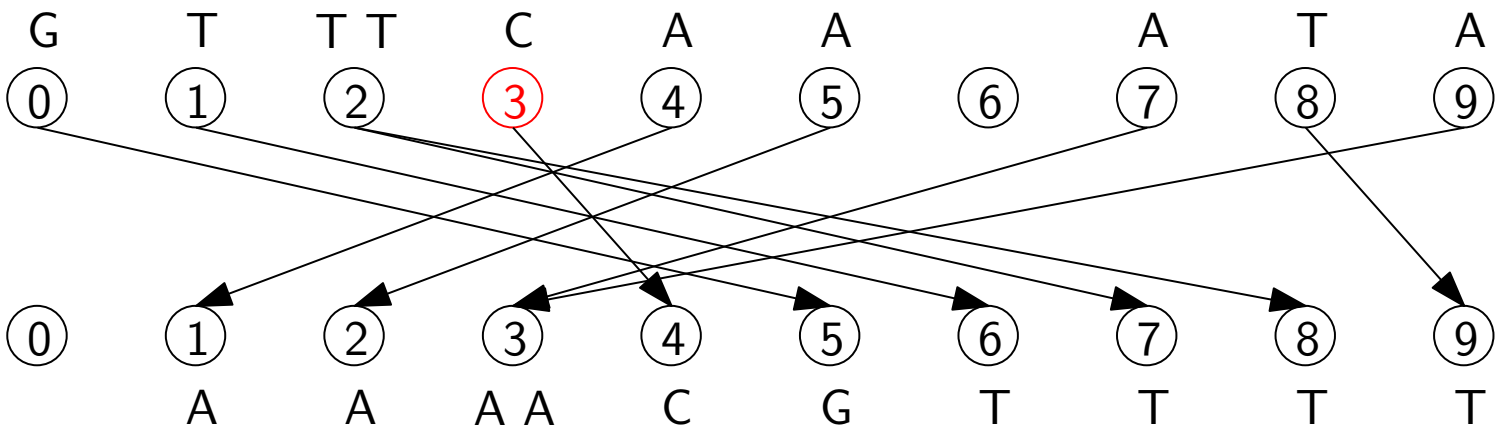
# seeking TA



# seeking TA



# *r*-indexing

# result

## Theorem

*Let $S$ be a string and $r$ be the number of runs in a BWT of $S$. We can store $S$ in $O(r)$ space such that later, given a pattern $P$, in $O(|P| \log\log |S|)$ time we can find the interval in the BWT containing characters preceding occurrences of $P$ in $S$, and then report the locations of those occurrences in $O(\log\log |S|)$ time per occurrence.*

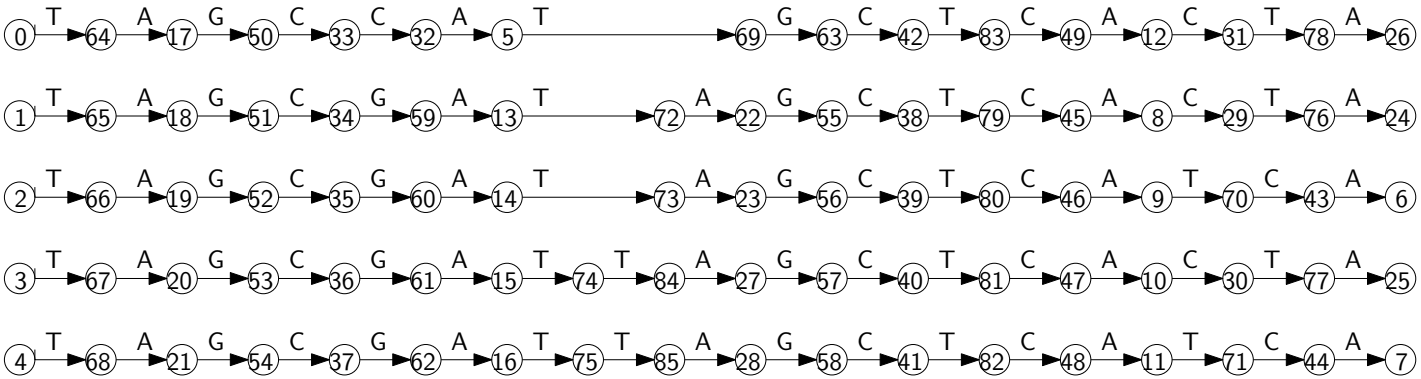
# result

## Theorem

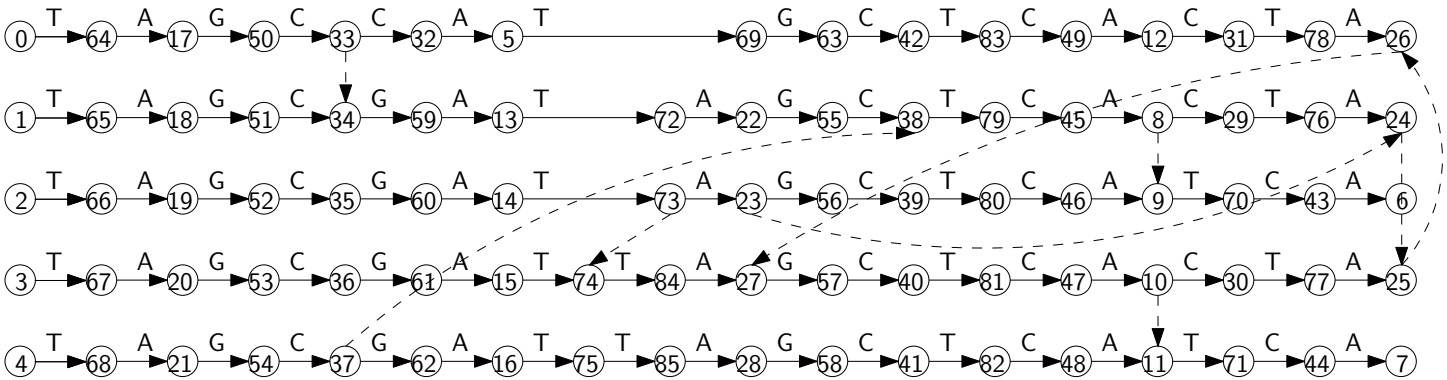
*Let $S$ be a string and $r$ be the number of runs in a BWT of $S$. We can store $S$ in $O(r)$ space such that later, given a pattern $P$, in $O(|P| \log\log |S|)$ time we can find the interval in the BWT containing characters preceding occurrences of $P$ in $S$, and then report the locations of those occurrences in $O(\log\log |S|)$ time per occurrence.*

Nishimoto & Tabei: total time is $O(|P| + \text{occ})$ when the alphabet size is polylog($|S|$).

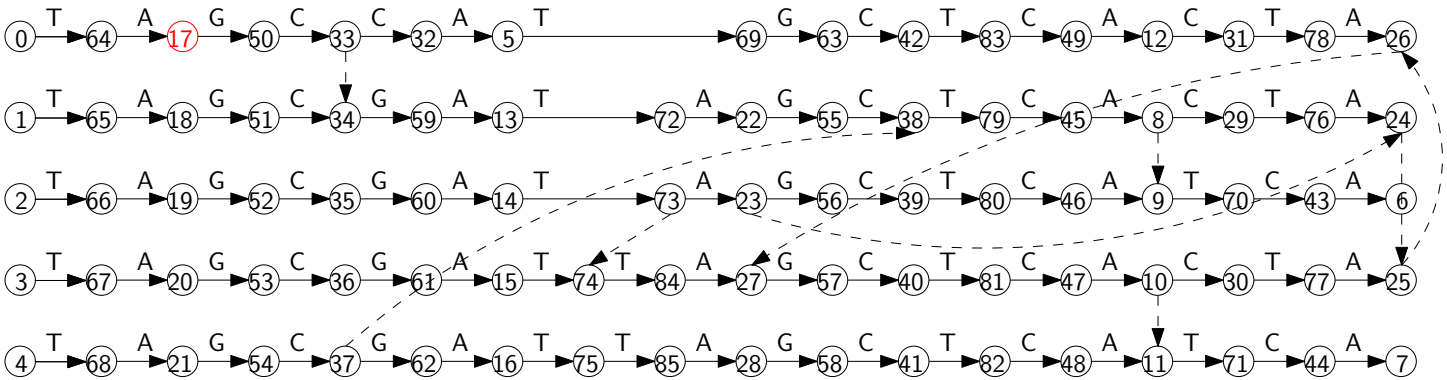
# seeking TA



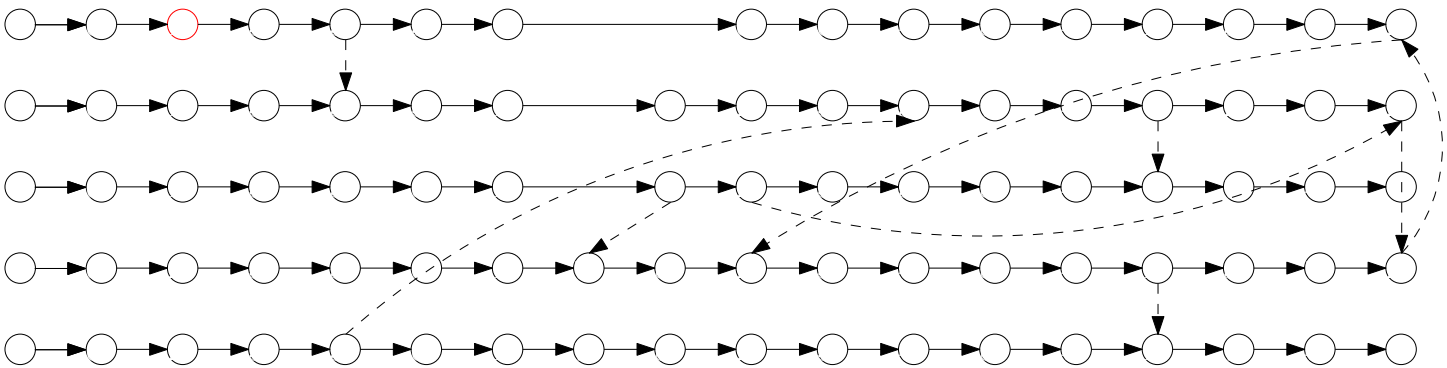
# seeking TA



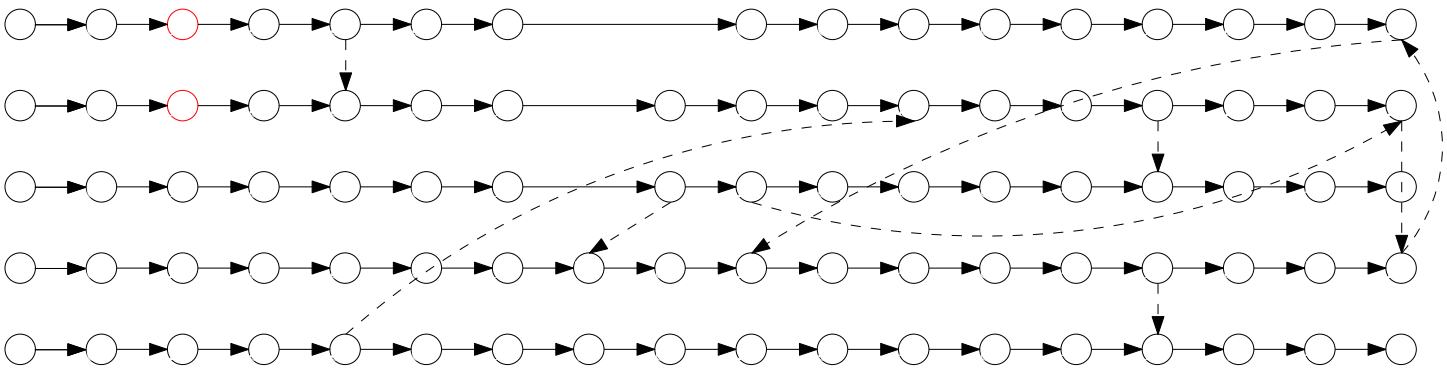
# seeking TA



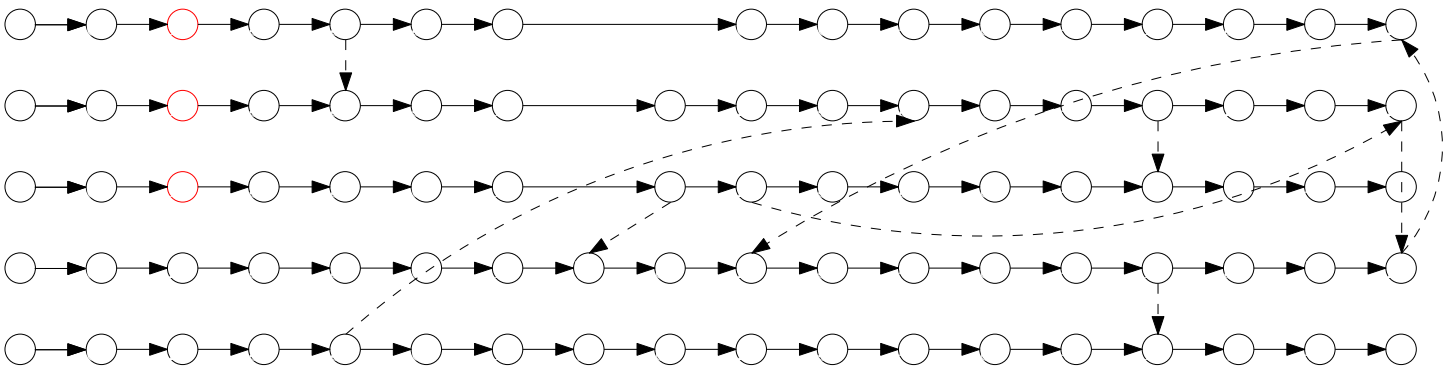
# seeking TA



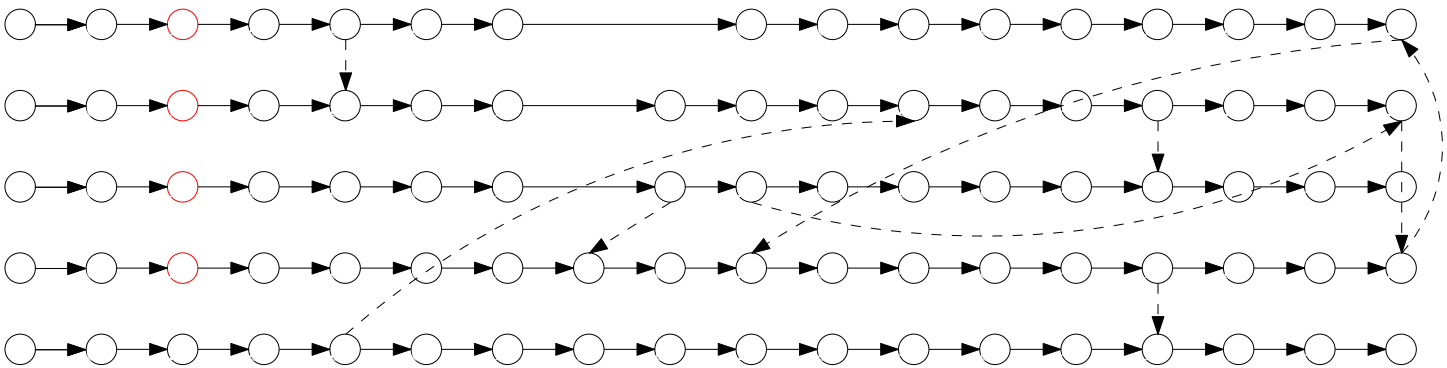
# seeking TA



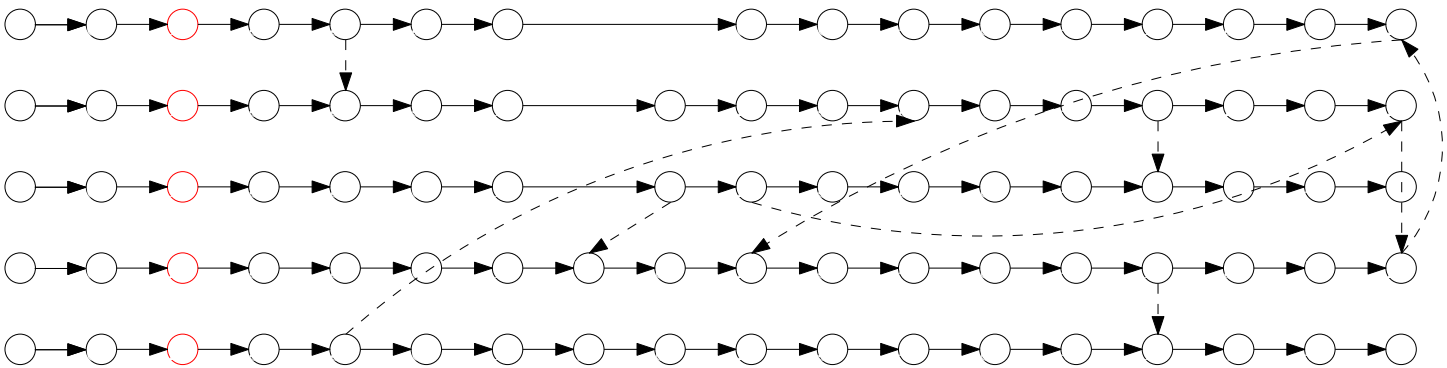
# seeking TA



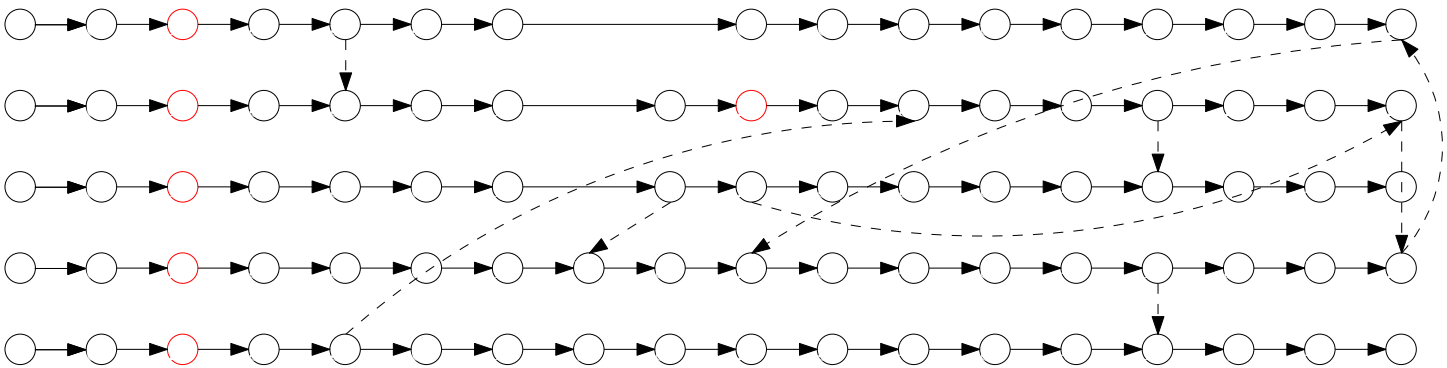
# seeking TA



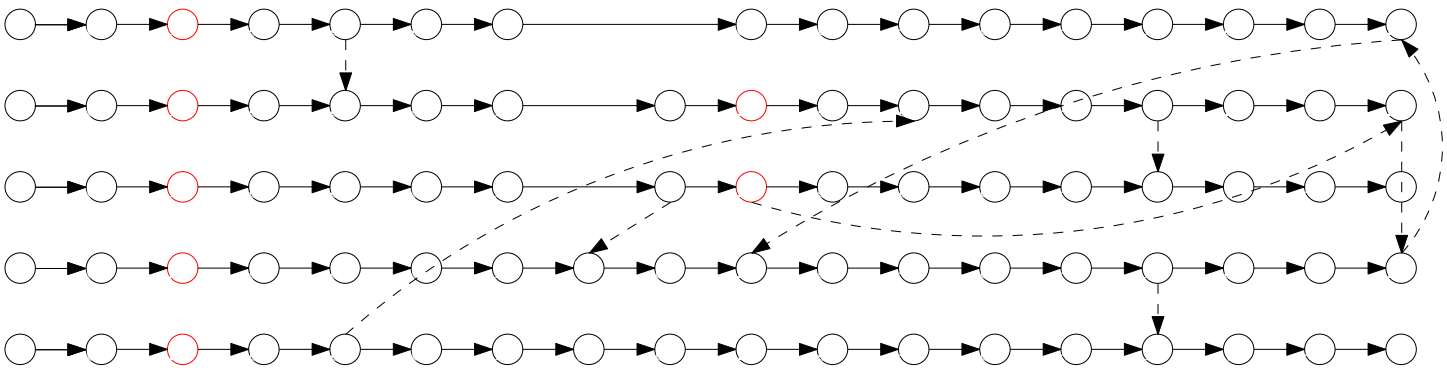
# seeking TA



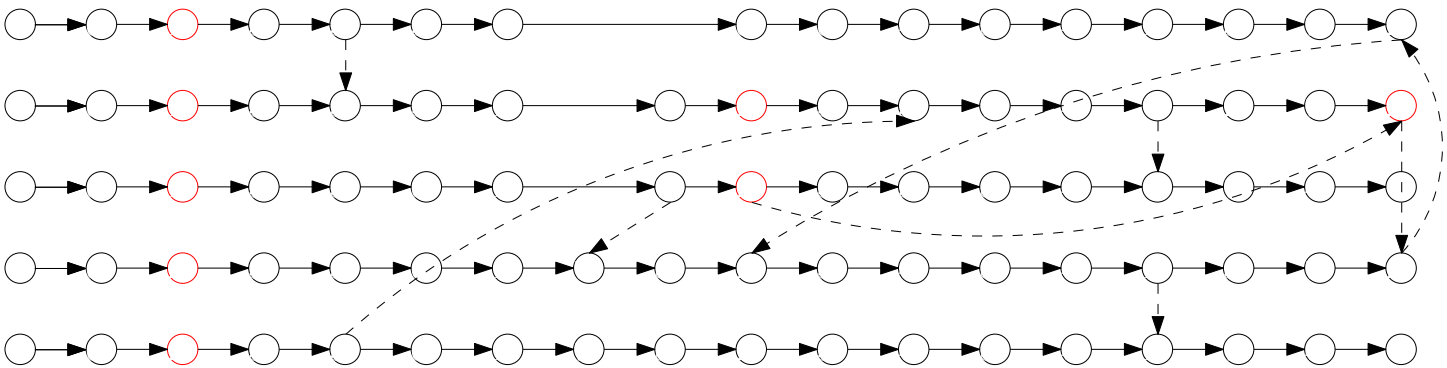
# seeking TA



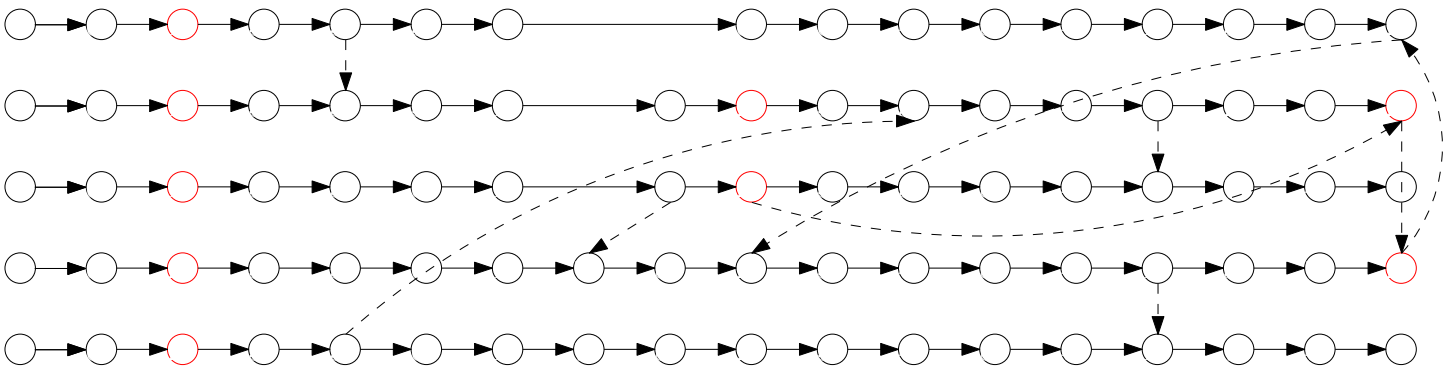
# seeking TA



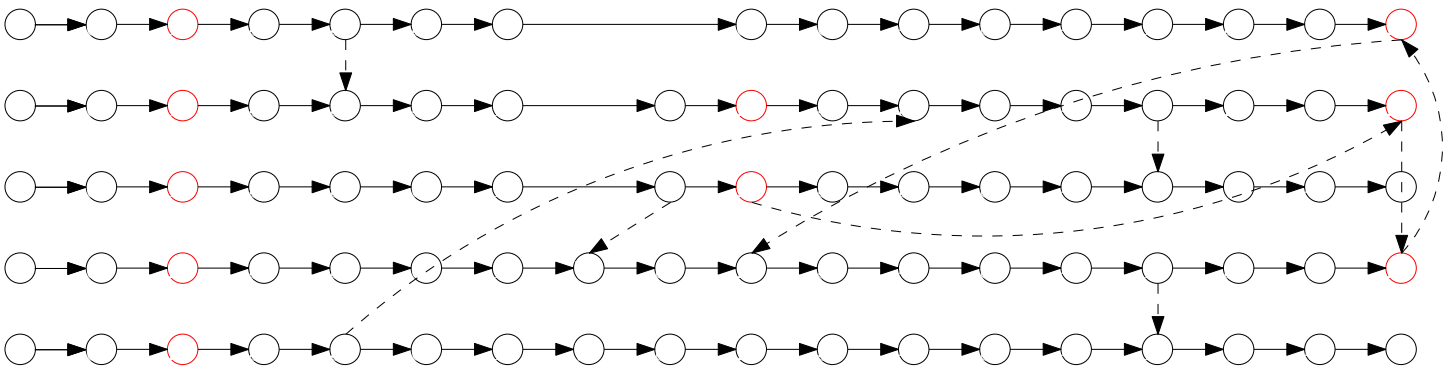
# seeking TA



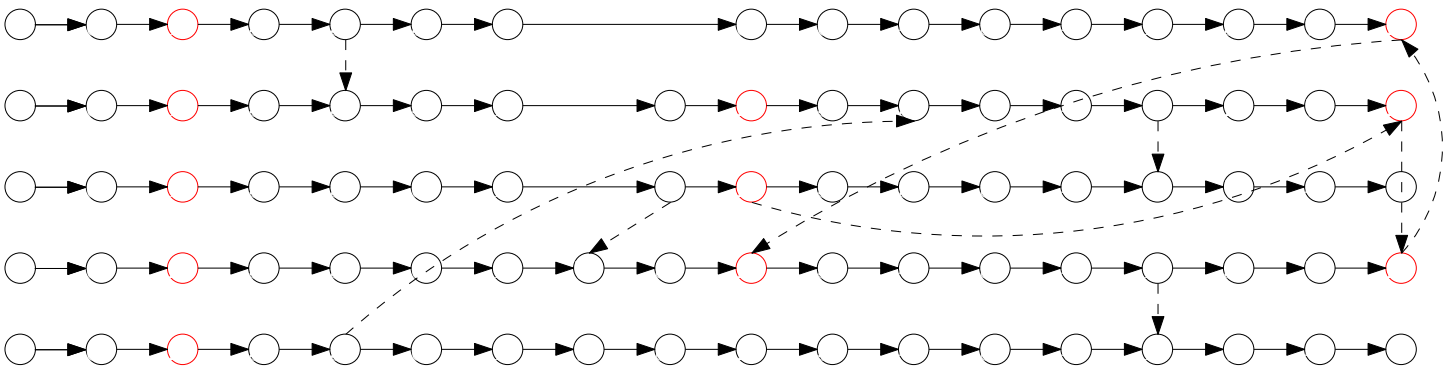
# seeking TA



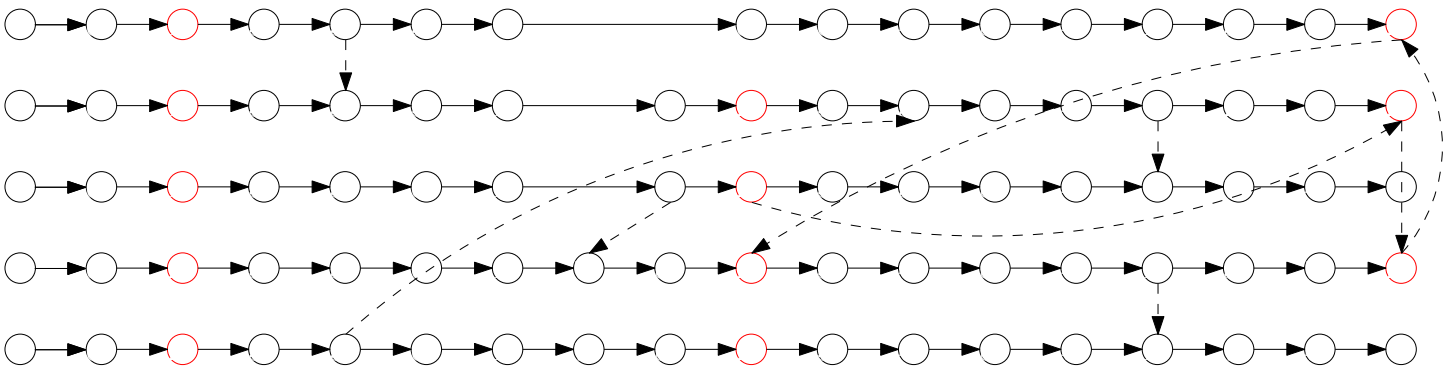
# seeking TA



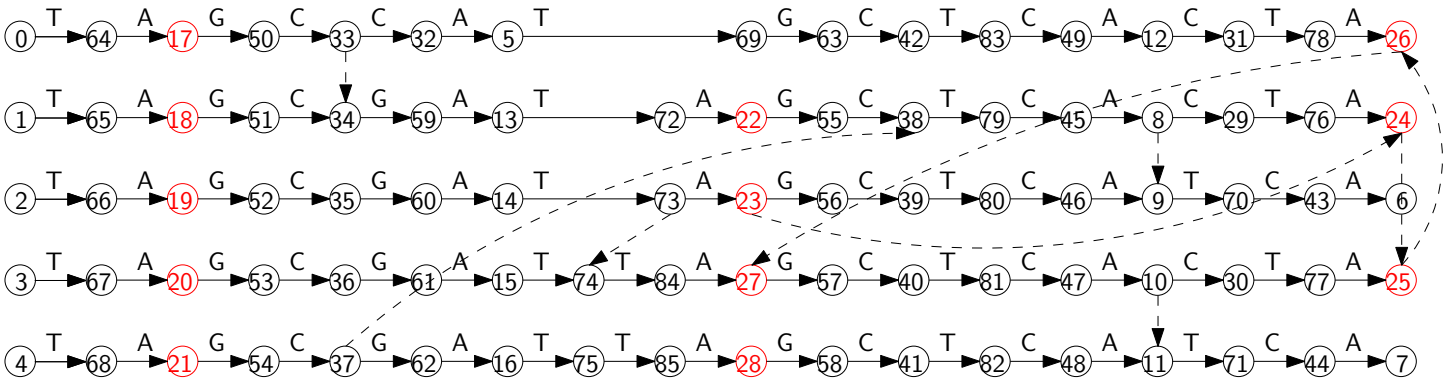
# seeking TA



# seeking TA



# seeking TA



# combination

# result

## Theorem

*Let $G$ be a Wheeler graph and $r$ be the number of runs in a BWT of $G$, and suppose $G$ can be decomposed into $\upsilon$ edge-disjoint directed paths whose internal vertices each have in- and out-degree exactly 1. We can store $G$ in $O(r + \upsilon)$ space such that later, given a pattern $P$, in $O(|P| \log\log |G|)$ time we can count the vertices of $G$ reachable by directed paths labelled $P$, and then report those vertices in $O(\log\log |G|)$ time per vertex.*

# result

## Theorem

*Let $G$ be a Wheeler graph and $r$ be the number of runs in a BWT of $G$, and suppose $G$ can be decomposed into $\upsilon$ edge-disjoint directed paths whose internal vertices each have in- and out-degree exactly 1. We can store $G$ in $O(r + \upsilon)$ space such that later, given a pattern $P$, in $O(|P| \log\log |G|)$ time we can count the vertices of $G$ reachable by directed paths labelled $P$, and then report those vertices in $O(\log\log |G|)$ time per vertex.*

Can we use Nishimoto & Tabei's technique to get $O(|P| + \text{occ})$ time when the alphabet size is polylog($|G|$)?

# mnemonics

1. $\upsilon$ ("upsilon") is the number of unitigs
2. Travis says \ups ("oops") a lot

# motivation?

We needed a special case of this theorem in a recent paper Garance will present, about long and stringy trees.

It's easier to prove the general theorem than to explain why we'd want it. Wheeler graphs are cool and *r*-indexes are cool, so *r*-indexing Wheeler graphs is really cool?

Jouni says he's already implemented something like this, so it must be useful for something...

Nicola has a more sophisticated result in a SODA '21 paper (but I'm not sure it's practical).

# tunnels

# definition

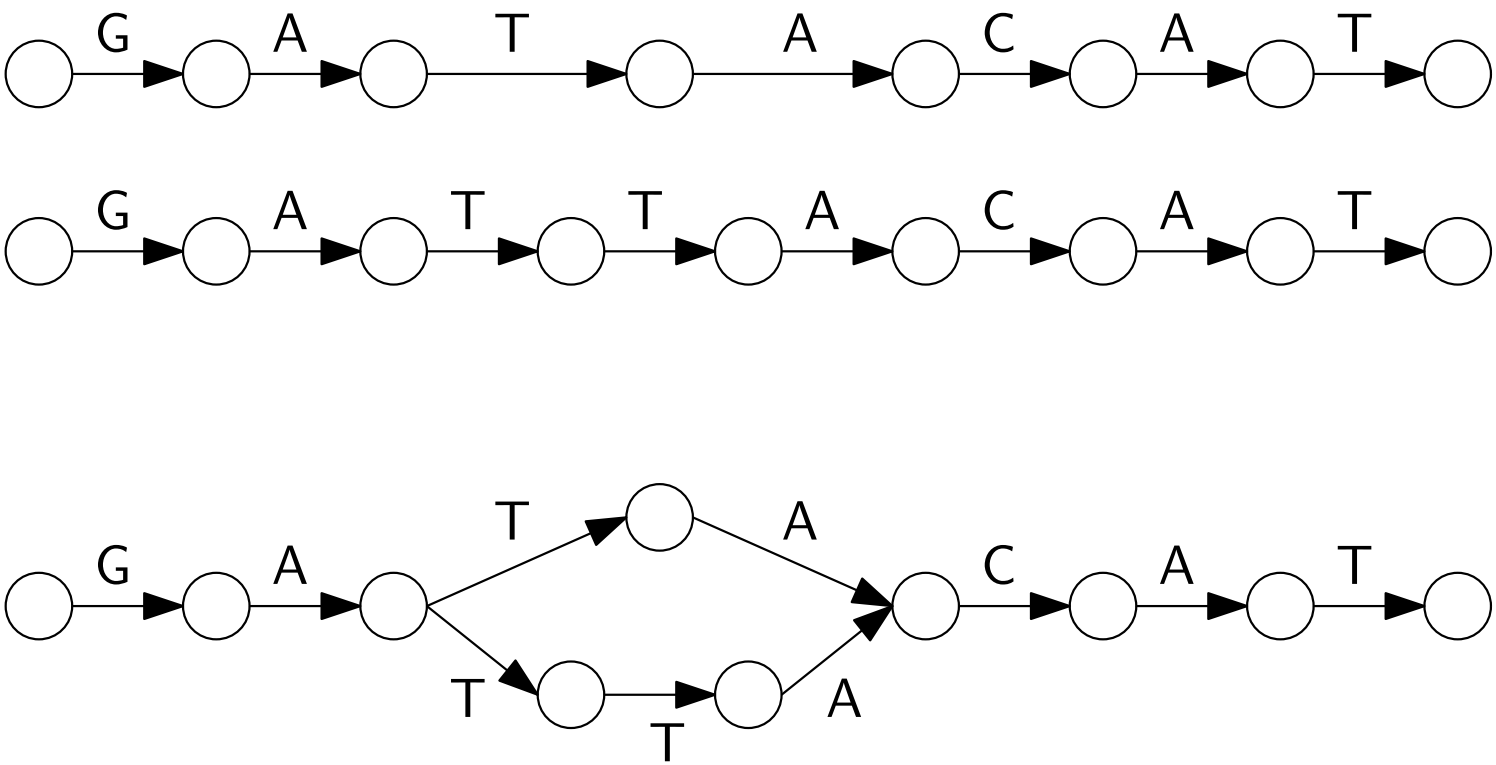
Uwe Baier introduced tunnelling to reduce the number of times we write large run-lengths in the RLBWT.

We can put paths in a Wheeler graph together in a tunnel if

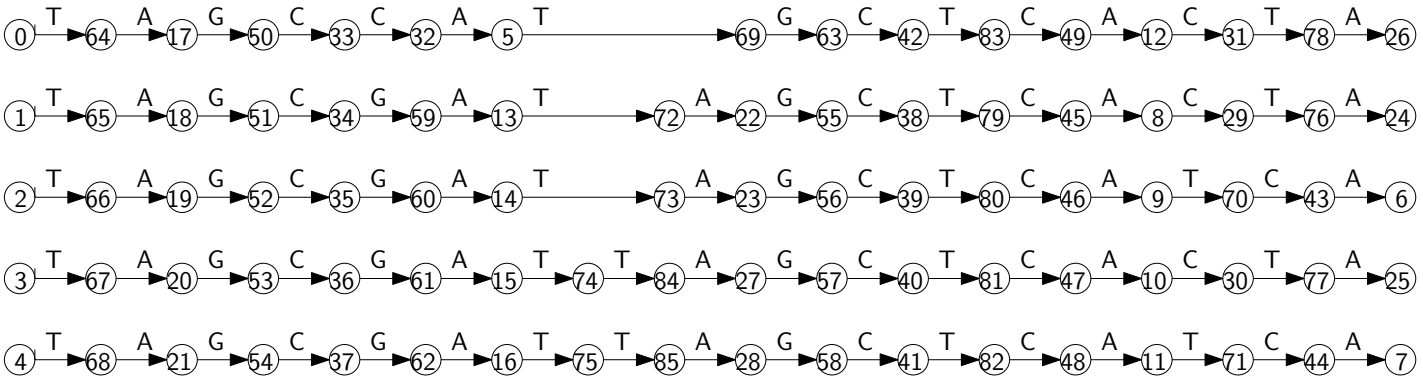
- corresponding edge labels match
- corresponding vertices' ranks are incremented in the Wheeler order.

We stop merging "one edge too early".

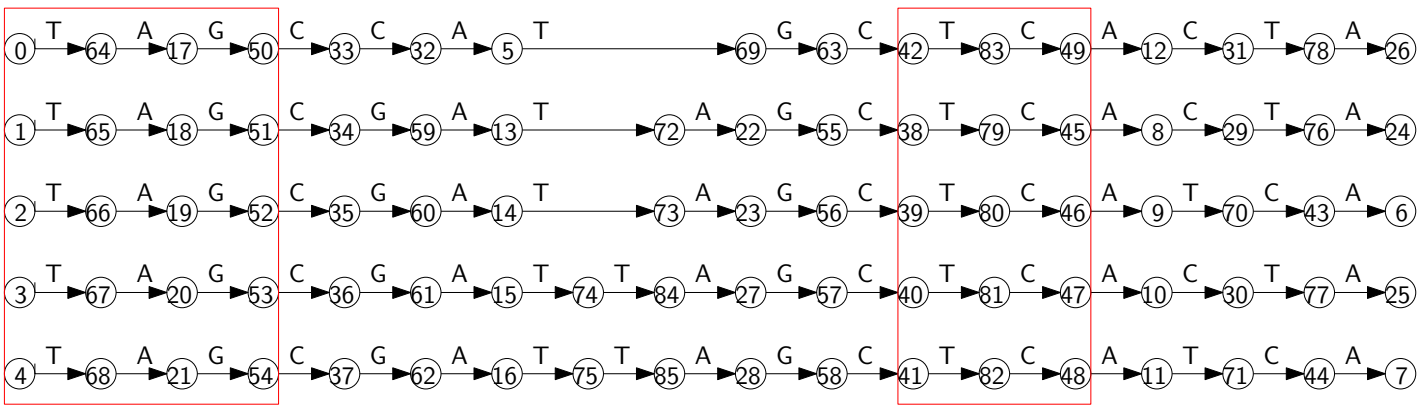
# examples



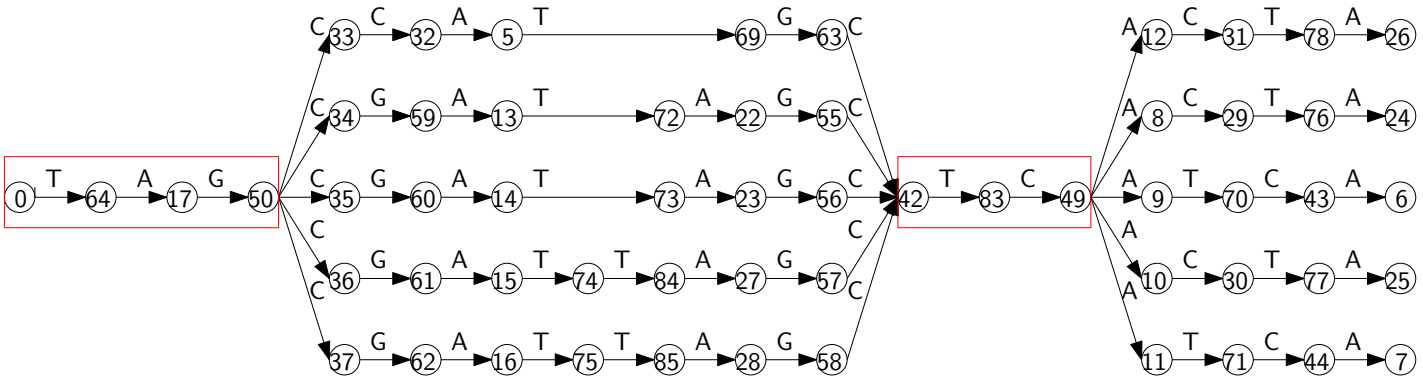
# examples



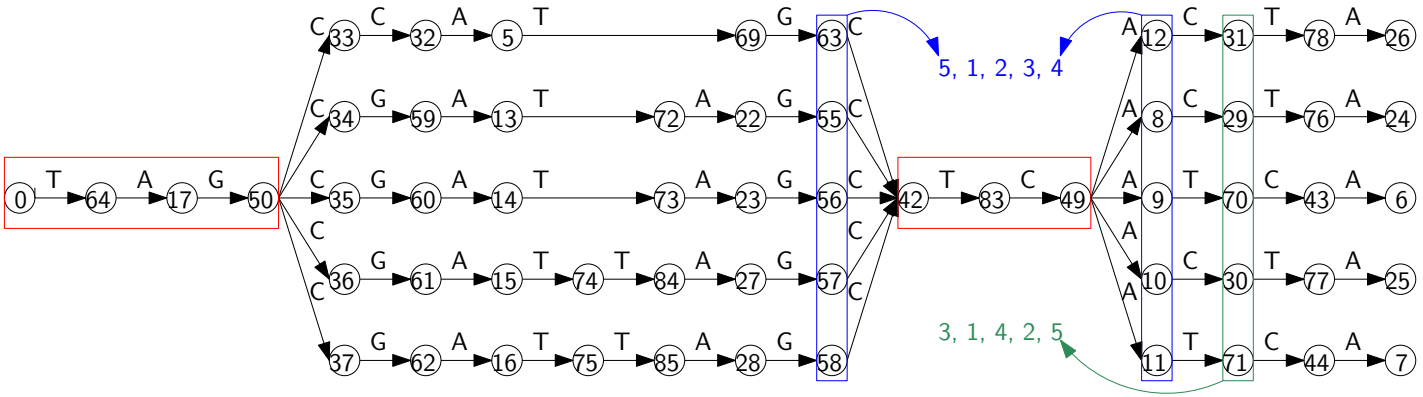
# examples



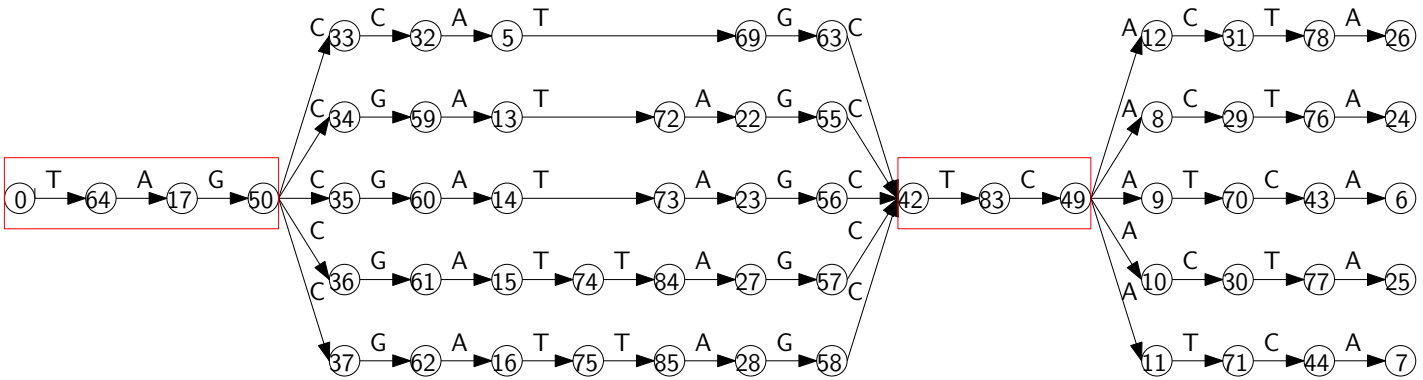
# examples



# examples



# chimeric matches




## Theorem

*We can index a tunnelled Wheeler graph so as to* prevent *false-positive, chimeric matches.*

(In our example, if the first variation is CAT, the second variation *isn't* TC.)

# mnemonics

1. 
2. tunnels are FIFO

# hand-waving

# doubt

Would we want to *r*-index a tunnelled Wheeler graph?

Doesn't tunnelling the graph collapse the runs in the BWT?

Wouldn't a standard sampling be better then?

# speculation

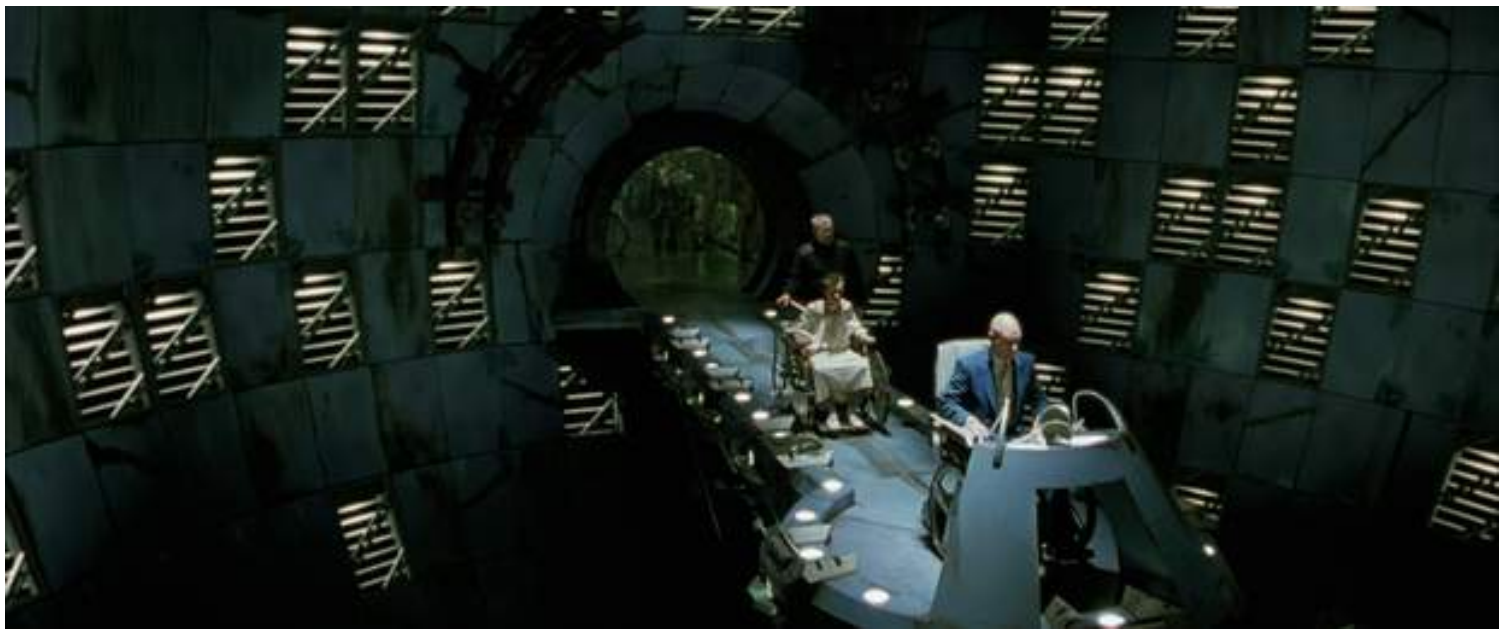
Suppose

- the conserved regions between variation sites are on the order of 1K bp
- the probability of a bp mutating is on order of 1 in 100M[†]
- we're indexing 1M genomes.

We'd like to tunnel the conserved regions, but how do we deal with the mutations?

[†]I've read this is the mutation rate in humans, but that seems to be the probability a bp is mutated between generations, not the probability of picking a bp and a person at random and having that bp mutated in that person. Any idea?

# speculation



(not recommended!)

# hope?

The "conserved" regions of about 1K bp are each broken up by about $(1\text{M} \times 1\text{K}/100\text{M}) = 10$ mutations, in expectation.

If we merge the paths in each conserved region into about 10 tunnels, however, then we *should* get a significantly smaller, interesting Wheeler graph whose BWT is still highly run-length compressible?

# questions?

Happy Year of the Ox!

Un saludo pa' Moni!